المدرسة الوطنية للعلوم التطبيقية
ⵜⵉⵎⵙⵙⵓⵜ ⵜⴰⵏⴰⵟⵓⵔⵉⵜ ⵜⵉⵣⵔⴰⵢⵉⵜ ⵜⵓⵏⵉⵣⵓⵉⵜ
Ecole Nationale des Sciences Appliquées de Fès

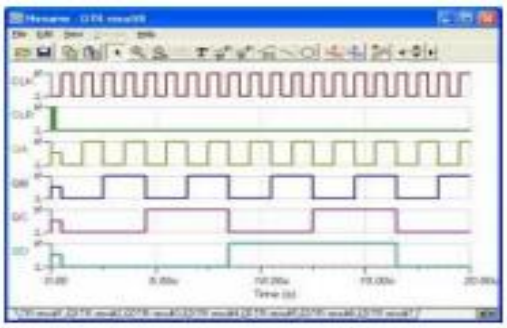
جامعة سيدي محمد بن عبد الله
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵓⵣⵍⴰⵣ ⵎⵓⵃⴰⵎⴷ ⵜⵉ ⵄⴱⴷⵍⵍⴰⵀ
Université Sidi Mohamed Ben Abdellah

1ère année GSEII

Circuits reprogrammables et VHDL

PROJET D'EVALUATION DU COURS VHDL & CIRCUITS REPROGRAMMABLES

Calculateur de vitesse de rotation d'un moteur par comptage d'impulsion



Réalisé par : - MIKOU Badr.

Encadré par : - Pr. MANSOURI Anass

Sommaire :

INTRODUCTION :

Ce projet est un **« calculateur de vitesse de rotation d'un moteur »** qui a comme objectif la détermination de la vitesse d'un moteur par comptage d'impulsions. Un capteur monté sur le stator du moteur émet des impulsions. Au que les impulsions soit calculées, le système envoie une valeur qui correspond à la vitesse du moteur.

Ce projet nous permet de mettre en pratique nos connaissances acquises pendant le cours de circuits reprogrammables et VHDL de la réalisation du circuit (description en VHDL) jusqu'à la simulation et la synthèse sur un FPGA (RTLViewer).

I- Contexte général du projet :

a- Contexte :

Un capteur actif (un capteur de proximité avec une électronique intégrée), alimentée avec une tension définie. Un anneau multipolaire, également inséré dans la bague d'étanchéité d'un roulement de roue, peut par exemple être utilisé comme roue d'impulsion (dans notre cas elle émet 30 impulsions). Dans cette bague d'étanchéité sont placés des aimants dont la polarité est alternée (dans notre cas 30 **N** et 30 **S**) (figure 1). Les résistances magnéto résistives intégrées dans le circuit électronique du capteur détectent le champ magnétique alternant lors de la rotation de l'anneau multipolaire. Ce signal sinusoïdal est transformé en un signal numérique par l'électronique dans le capteur (figure 2). La transmission vers le calculateur se fait par un signal électrique avec le procédé de modulation de largeur d'impulsions.

Le capteur est relié à un **calculateur** qui calcule le nombre d'impulsion captées, en même temps un autre calculateur appelé **« calculateur de vitesse »** calcule le nombre de cycle d'horloge nécessaire pour que la roue termine un toure. Après que la roue termine son tour le nombre de cycle d'horloge est envoyé comme sortie.

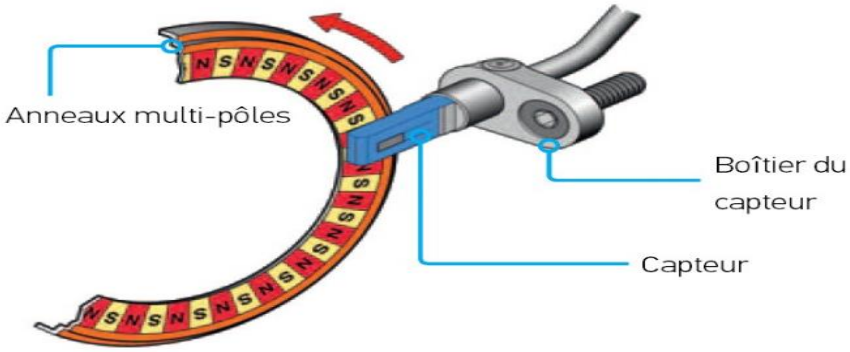


Figure 1 : Anneaux-multipôles et capteur.

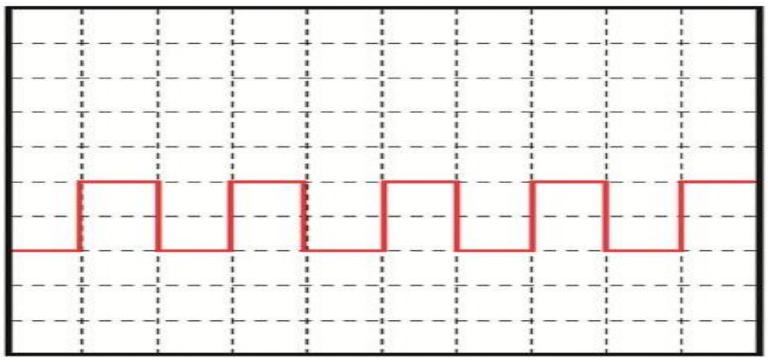


Figure 2 : Signal capté et numérisé par le capteur

b- Cahier de charge :

Le circuit proposé dans ce projet est un **« calculateur de vitesse de rotation d'un moteur »** qui a pour objectif de traiter les signaux émis par le capteur en but de déterminer la vitesse de rotation. Pour réaliser ce système, on a décidé de créer un compteur qui, à chaque fois que le moteur fait un tour, il envoie sa valeur sur la sortie. Le compteur est incrémenté à chaque front d'horloge.

- **Les Entrées/Sorties :**

➔ Les entrées :

- i_in : entrée provenant du moteur sur laquelle on peut relever les impulsions correspondant aux tours du moteur. Impulsions comptées sur front montant.
- rq : ligne provenant d'un système externe informant le composant qu'une valeur de la vitesse veut être lue.
- clk : l'horloge du système cadencée à 50 MHz
- rst : Remise à zéro du composant. Actif à l'état bas.

➔ Les sorties :

- v : représente la vitesse du moteur codé sur 21 bits.
- seg_i : se sont les sorties 7 segment qui permettent de visualiser la valeur de v sur des des afficheur 7 segment (i : 1 → 6).
- ack : le circuit informe l'extérieur que la vitesse peut être lu.

- **Les Objectifs/Contraintes :**

➔ Les objectifs :

Vitesse minimale :	Vitesse maximale :
La vitesse minimale souhaité est de 12 tours/s ce qui veut dire 1 tour chaque 83 millisecondes : Si on fait un calcule : - L'horloge sur 25 MHz. - $2^{21} - 1 = 2\,097\,151$(tous les bits de v reçoivent 1) (2 097 151 cycles d'horloge pour un tour) ➔ $2\,097\,151 / (25 \times 10^6) = 0.083s = 83$ ms	La vitesse maximale quand peut calculer est de : - La durée d'une impulsion est au moins 40 ns. Et la durée qui les sépare doit aussi être au moins 40 ns. (Voir les contraintes). ➔ 30 x 80 = 1tour dans 2400 ns ➔ 416 666 tours/s Cela veut dire que la fréquence d'horloge est suffisante pour un moteur très rapide.

➔ Les contraintes :

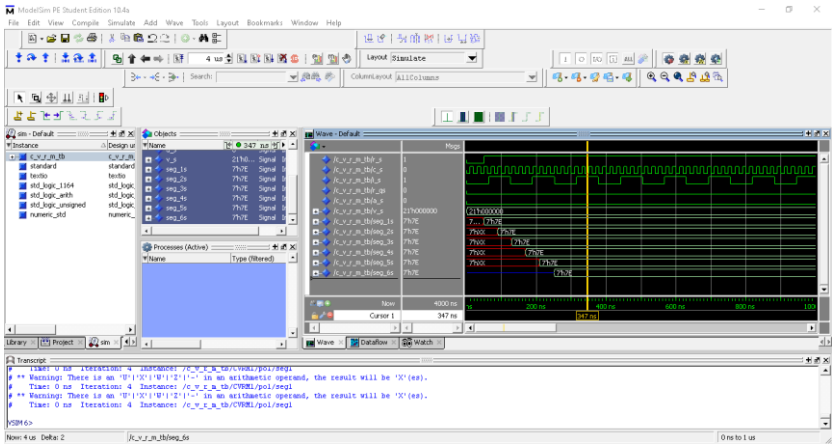
- La contrainte de l'horloge : plus la fréquence de l'horloge est grande plus on aura besoin de plus de bits pour la sortie.
- La relation entre impulsion est l'horloge : les impulsions doivent avoir une taille supérieure ou égale au cycle d'horloge pour qu'on soit certain qu'elle soit détectée. Aussi la séparation entre les impulsions doit être supérieure ou égale au cycle d'horloge pour qu'elle soit tous détectées.
- Si l'impulsion est détectée plusieurs fois par l'horloge alors elle sera calculée plusieurs fois.

II- Conception et Modélisation du projet :

a- L'environnement technique du projet :

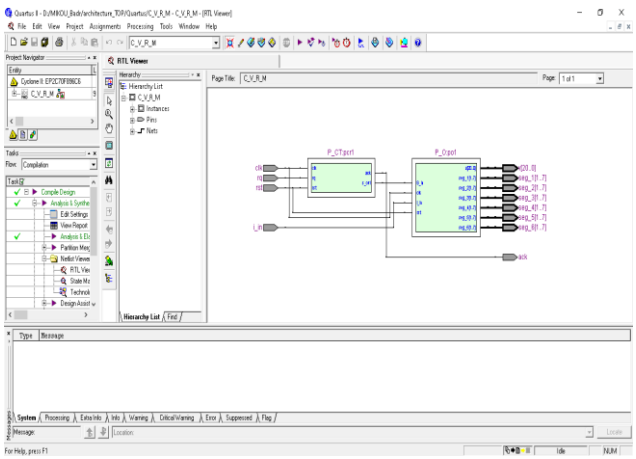
- Logiciel de conception et de simulation :

- Le logiciel utiliser pour la conception des circuits que cela soit des circuits de base ou des sous-blocks ou les blocks principaux ou l'architecture TOP est : **MODELSIM PE STUDENT EDITION**.
- La conception de chaque circuit se fait dans un **dossier qui porte son nom** qui est lui-même divisé en quatre sous dossiers :
  - ➔ Le dossier **modelsim :** dans le il existe un scripte de compilation et de simulation avec une extension **« .do »** et la bibliothèque **work** qui notre environnement de compilation et de simulation (le dossier du circuit doit être dans le disque dur D pour le fonctionnement du script).
  - ➔ Le dossier **SRC** : dans lequel il y a la **description VHDL** du circuit
  - ➔ Le dossier **test_bench** : on y trouve la description du teste du circuit (la simulation).
  - ➔ Le dossier **Quartus** : dans lequel il y a la **synthèse** sur un **FPGA**.



- Logiciel de synthèse :

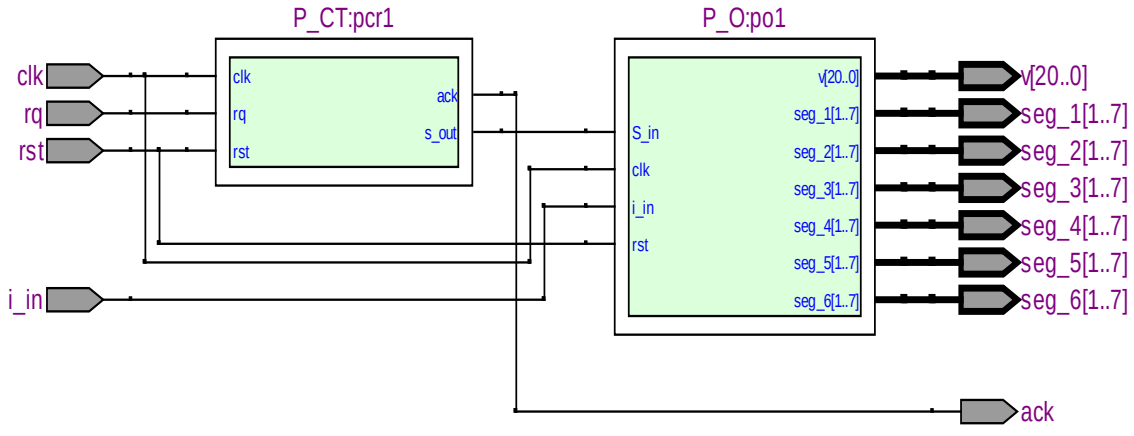
- Le logiciel utiliser pour la synthèse est QUARTUS II 9.0 (32 bits).
- La carte utilisé pour la synthèse et une carte FPGA de la famille Cyclone II et de code EP2C70F896C6



b- Architecture générale du circuit :

Notre système est composé de deux partie :

- Une partie contrôle, son rôle principal est de communiquer la vitesse de rotation du moteur à l'autre système extérieur Elle a 3 entrées et 2 sorties :
  - ➔ rst : pour la remise à 0, active bas.
  - ➔ clk : représente l'horloge.
  - ➔ rq : représente la demande de lecture de la vitesse (request).
  - ➔ ack : c'est une sortie sont but est de déclarer que la vitesse peut être lue (acknoledge).
  - ➔ s_out : c'est ordre envers la partie opérative pour but de fixer la vitesse sur la sortie de la partie opérative (stop).
- Une partie opérative c'est la partie puissante de notre machine, elle est composée de plusieurs sous blocks, son rôle est de calculer la vitesse de rotation en attendant l'ordre de fixation de la vitesse sur les sorties, venu de la partie contrôle. Cette partie est composé de 4 entrées et de 7 sorties :
  - ➔ rst : pour la remise à 0, active bas.
  - ➔ clk : représente l'horloge.
  - ➔ i_in : représente les impulsions envoyées par le capteur.
  - ➔ s_in : représente l'ordre donné par la partie contrôle.
  - ➔ v : une sortie sur 21 bits représente la vitesse.
  - ➔ seg_i : (i : 1 → 6) des sortie sur 7 bits représentent le codage de 4 bits de la vitesse avec un codeur 7 segments. Chaque i représente la case de l'octet.



c- Architecture détaillée du circuit :

- La partie opérative (partie_OP ou P_O) :

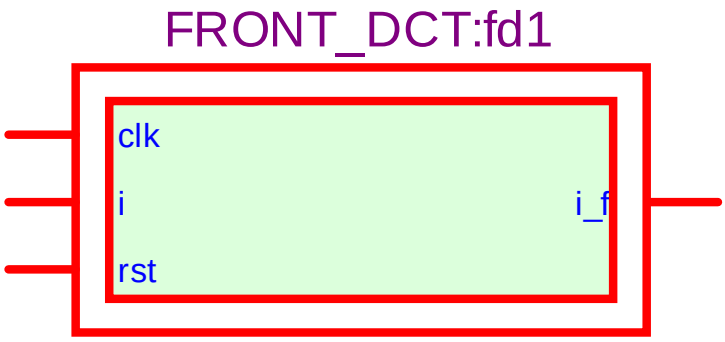
La partie opérative est composé de plusieurs sous-blocks :

➔ Diviseur de fréquence :

Elle comme entré l'horloge **clk** et le bit de remise à zéro **rst** et une sortie sur 1 bit qui est **cout**. D'avoir un signal périodique de fréquence F2 grâce à un signal périodique de fréquence F1 tel que la fréquence F2 et inférieur à F1.

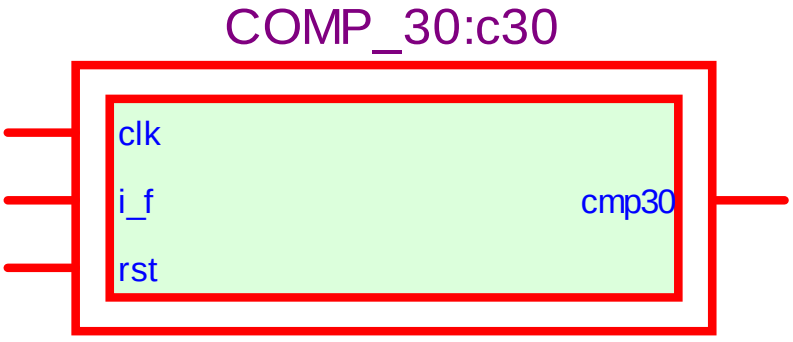
➔ **Détecteur de front :**

Son rôle est de comparer deux instants successifs par rapport à l'horloge avec l'équation **S = A AND NOT(B)** afin de voir si l'état est passé de bas vers le haut .



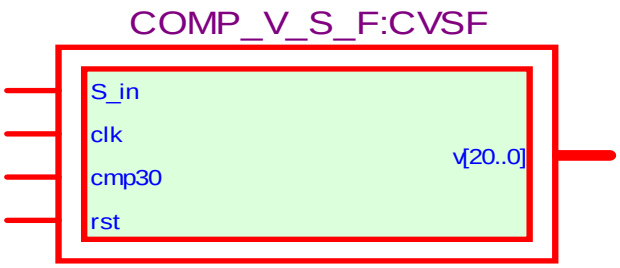
➔ **Compteur modulo 30 :**

Le compteur modulo 30 a pour rôle de compter de 0 jusqu'à 29. Quand il atteint 29 iles remis à zéro el il envoie une impulsion de fin de comptage.



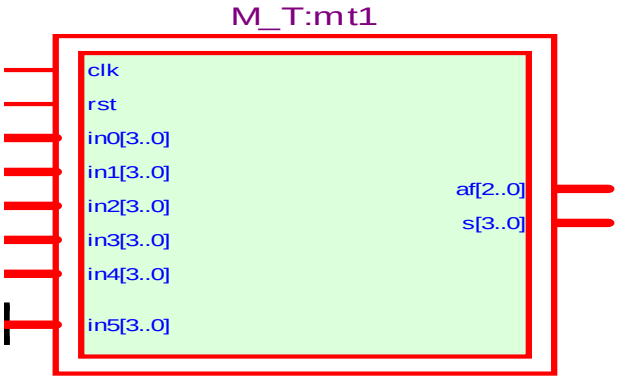
➔ **Compteur de vitesse, sauvegarde et fixation :**

Le compteur de vitesse est un simple compteur qui s'incrémente à chaque top d'horloge et qui est remis à zéro. Ensuite, il y a un étage de sauvegarde de la valeur comptée. Enfin, l'étage de fixation permet de fixer la valeur de sortie pendant que le système externe la lit.



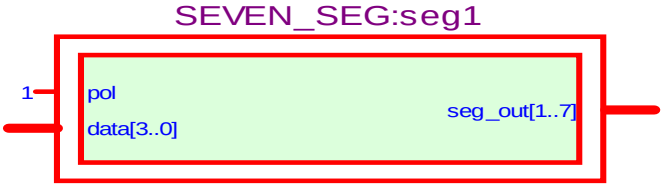
➔ Multiplexeur temporel :

C'est un multiplexeur qui a chaque cycle d'horloge envoie une des entrées à la sortie, ainsi que le numéro de sans classement ou sa case.



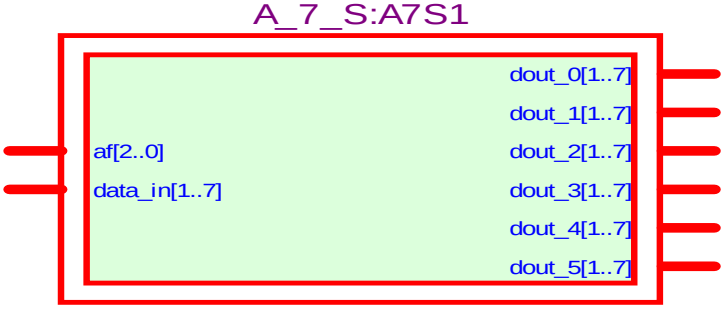
➔ Décodeur7 segment :

Son rôle est de décodé un octet à un segment de 7 bits affichable dans un afficheur 7 segments.



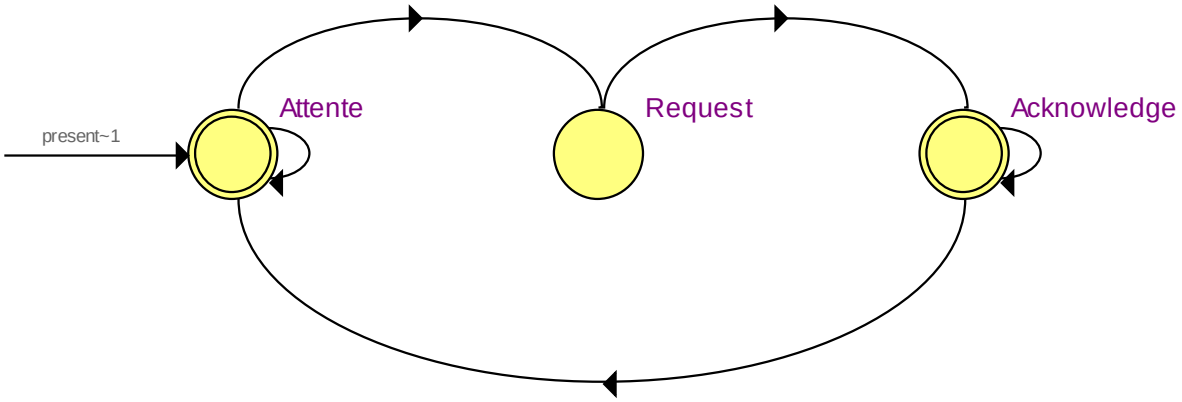
➔ Afficheur 7 segment :

Permet d'afficher chaque entrée dans sa case qui est donné a l'entré.



- **La partie contrôle (partie_CT ou P_CT) :**

Pour communiquer notre vitesse **v** au système externe qui n'utilise pas la même horloge, on utilise un protocole de communication asynchrone nommé handshaking. Le fonctionnement de ce système est le suivant, lorsque le système externe veut lire la valeur de V, il envoie une requête en mettant la ligne rq (Request) du composant à l'état haut. Notre composant détecte cette requête et fixe la valeur de **v** pour s'assurer que cette valeur ne sera pas modifiée avant la fin de la lecture, puis envoie l'autorisation de lecture via la ligne ack (Acknowledge). Quand le système externe a fini de lire il remet la ligne Request à 0. Enfin lorsque notre composant a détecté la fin de la lecture, il remet Acknowledge à 0 pour indiquer qu'il est prêt à accepter une nouvelle requête.



III- Réalisation et teste du circuit :

a- La réalisation du circuit :

- Les circuits de base :

Le block opératif et composé de plusieurs sous-blocks, chaque sous-block es composé d'un nombre de circuits de base. Certain de c'est circuit sont normaux par contre d'autres son génériques (le nombre de bit d'entrées et de sortie n'est pas spécifié) cela pour éviter la reconception de chaque circuit de base pour chaque sous-block.

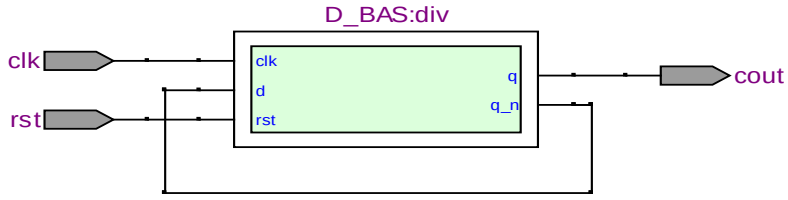
Les circuits de base utilisé sont :

Circuit non générique :	Circuit générique :
- Un circuit qui résout l'équation : ➔ S = A AND NOT(B) - Un comparateur. - Un multiplexeur 6 vers 1. - Une bascule D.	- Un additionneur de 1. - Un multiplexeur 2 vers 1. - Un registre flip- flop.

- Les sous-blocks :

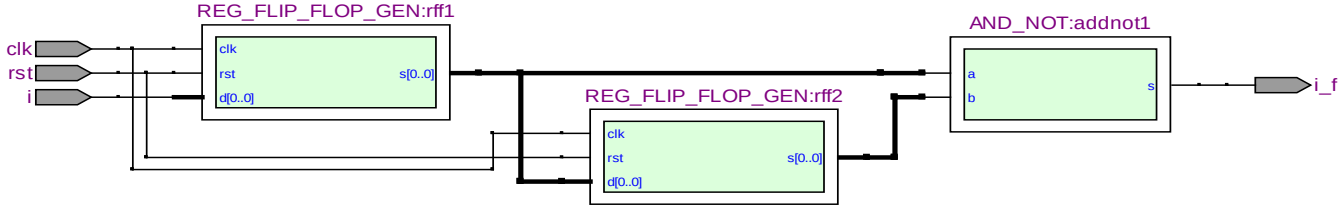
➔ Diviseur de fréquence :

Le rôle de ce diviseur dans ce système est d'avoir une de fréquence 25 MHz (utilisé par tous le système) à partir de clk qui a une la fréquence de 50 MHz (cas de l'horloge sur la carte FPGA). Cela à fin d'éviter le problème de grand nombre de bit qui représente la vitesse de rotation, ainsi pour avoir une durée importante pour le traitement de l'impulsion et le calcule. Il est composé d'une bascule D dont la sortie l'entrée d est liée à la sortie a bar. Il a aussi une entrée de remise à 0.



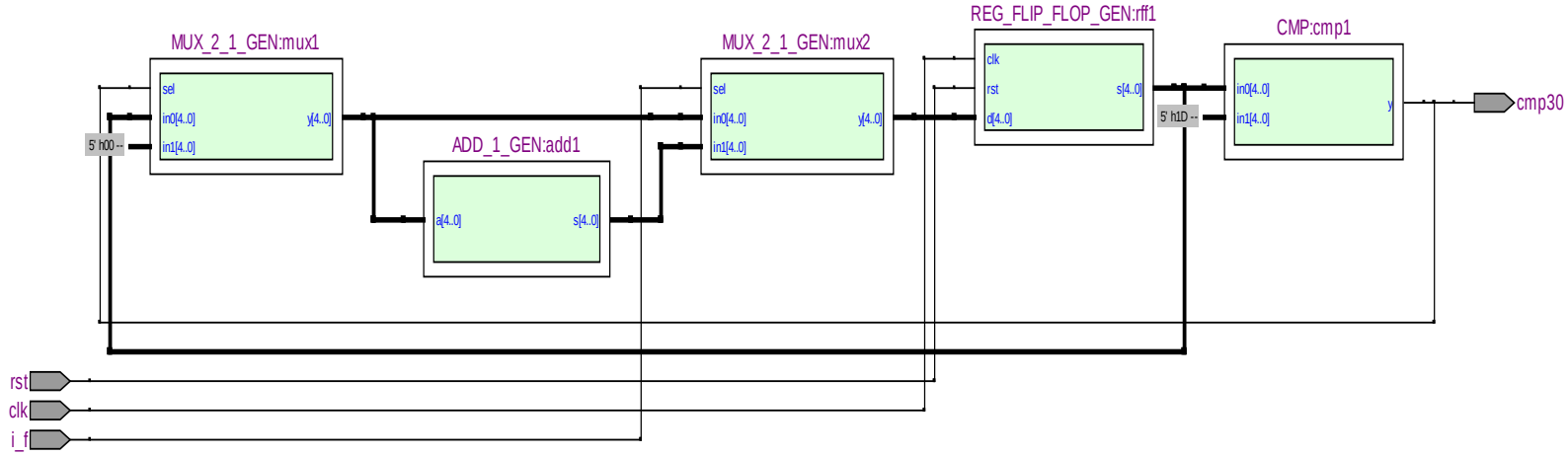
➔ Détecteur de front :

Son rôle est d'éviter de calculer l'impulsion entrante plusieurs fois (si la largeur de l'impulsion est supérieur à celle de l'horloge). Ce sous-blocks a comme entrées d'horloge, de remise à0 et aussi **l'impulsion** et il a comme sortie **i_f** qui passe 1 cycle d'horloge. Cette sortie est branchée directement vers le compteur modulo 30. Il est composé de deux registre flip-flop et le circuit 'ANDNOT'**.**



➔ Compteur modulo 30 :

Le compteur modulo 30 a pour rôle de compter 30 impulsions venant du moteur par l'intermédiaire de I_f. Une fois les 30 impulsions atteintes, le compteur génère un signal cmpt30 pendant une période de l'horloge qui indique au compteur de vitesse de sauvegarder sa valeur et de recommencer à zéro. Il est composé de deux multiplexeur s 2 vers 1 un additionneur de 1 un registre flip-flop et un comparateur qui Compart si le compteur a atteint 29 ou pas.

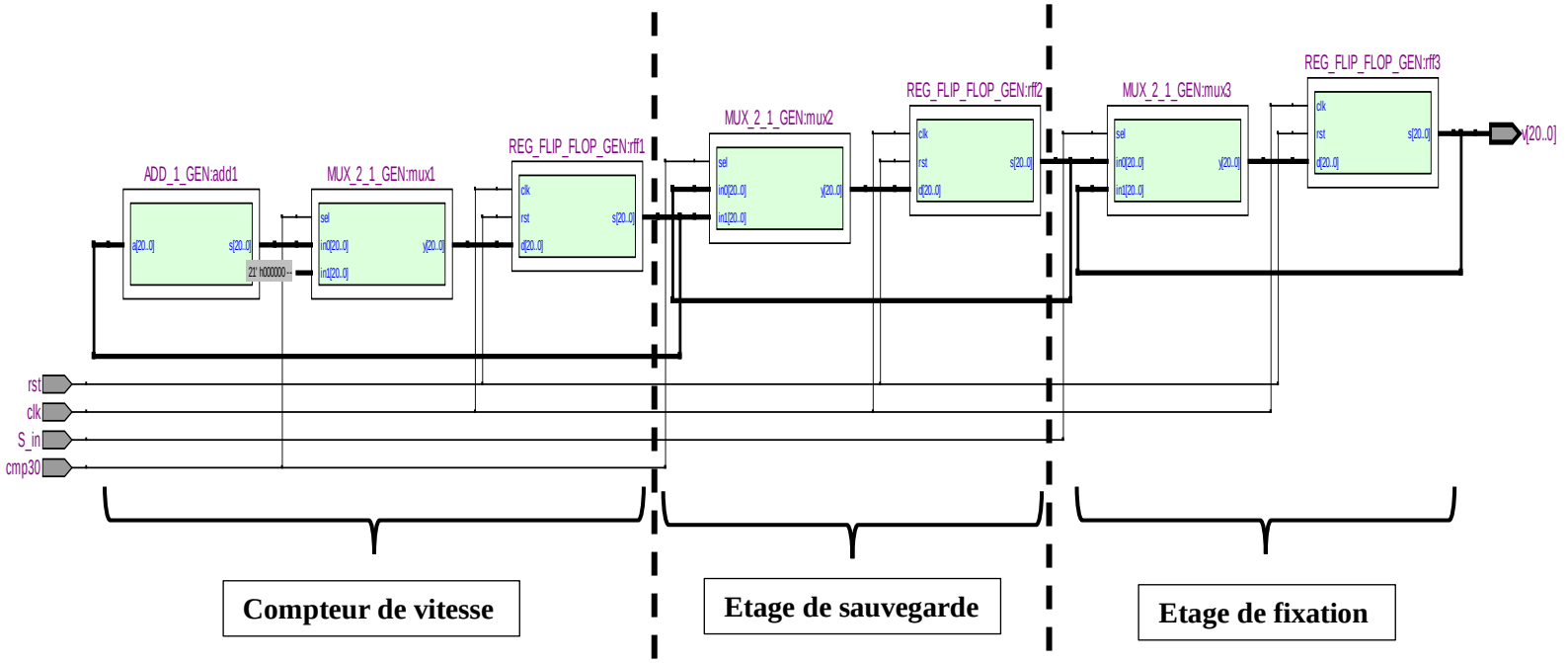


➔ Compteur de vitesse, sauvegarde et fixation :

Le compteur de vitesse est un simple compteur qui s'incrémente à chaque top d'horloge et qui est remis à zéro, soit par le raz global du circuit, soit par cmpt30 provenant du compteur modulo 30.

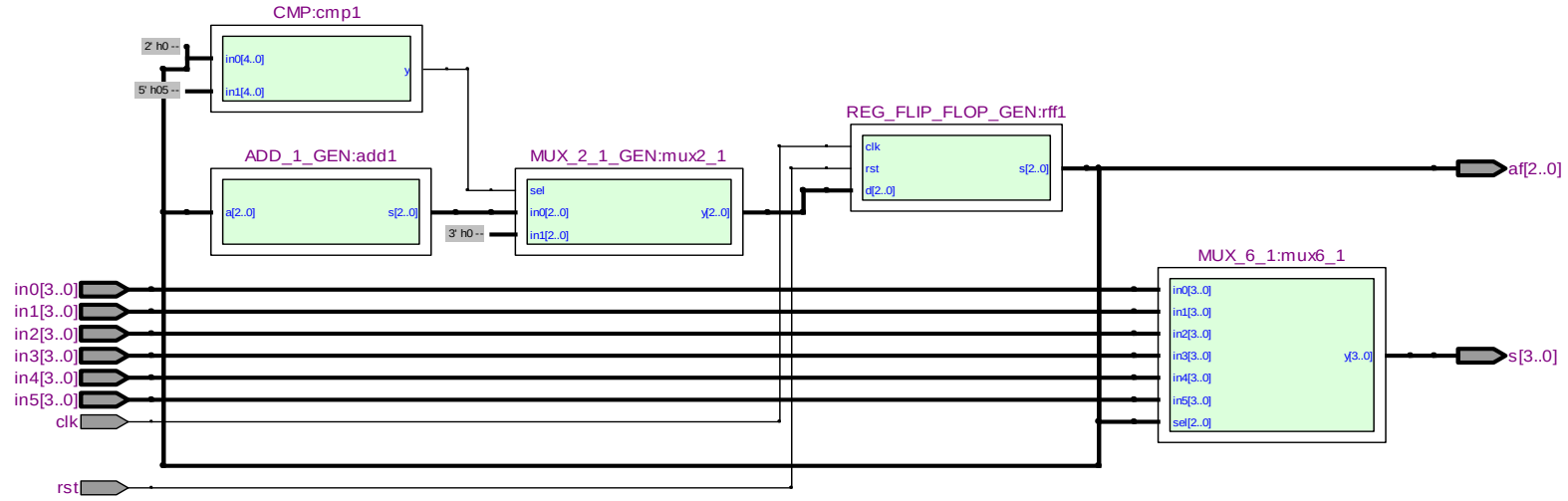
Puis il y a un étage de sauvegarde de la valeur de la vitesse à chaque fois que le moteur fait 1 tour. C'est un simple multiplexeur avec une bascule qui permet de faire cet étage pour chaque bit du registre.

Enfin, l'étage de fixation permet de fixer la valeur de sortie pendant que le système externe la lit. Cette fixation se fait grâce au signal Stop **s_in** qui est activé par la machine d'état.



➔ Multiplexeur temporel :

Dans ce circuit ce multiplexeur jour un rôle très important il nous évite d'utiliser plusieurs décodeur 7 segments à la fois en envoyant à chaque cycle d'horloge un octet de la vitesse vers le décodeur et sa case vers l'afficheur 7 segments. Les 21 bits sont divisés en 6 octets (le 21ème bit est inséré seul dans un octet) ce multiplexeur envoie à chaque cycle d'horloge 1 de ces 6 octets et sa case aussi. Il est composé d'un compteur qui compte jusqu'à 6 et un multiplexeur 6 vers 1, l'entrée de sélection est codée sur 3 bits.



➔ Décodeur 7 segment :

Il utilise le tableau suivant : (pour nous POL = 1).

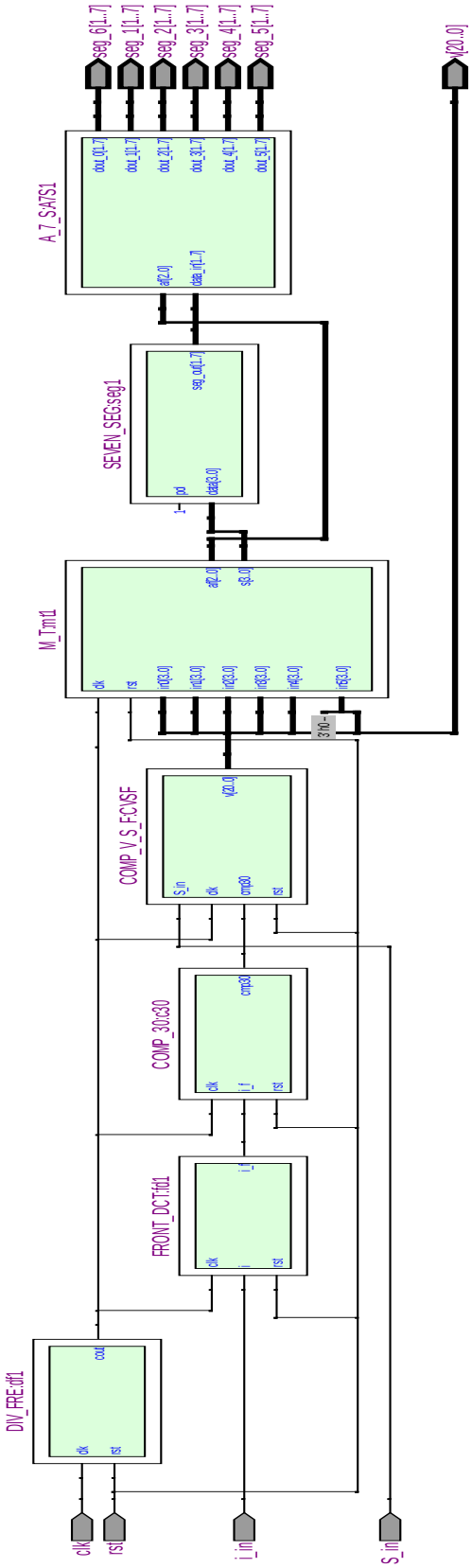
En hexadécimale :	En 7 Segment :	POL (active 0/1)
"0000" "0000" "0001" "0001" "0010" "0010" "0011" "0011" "0100" "0100" "0101" "0101" "0110" "0110" "0111" "0111" "1000" "1000" "1001" "1001" "1010" "1010" "1011" "1011" "1100" "1100" "1101" "1101" "1110" "1110" "1111" "1111"	"1111110" "0000001" "0110000" "1001111" "1101101" "0010010" "1111001" "0000110" "0110011" "1001100" "1011011" "0100100" "1011111" "0100000" "1110000" "0001111" "1111111" "0000000" "1111011" "0000100" "1110111" "0001000" "0011111" "1100000" "1101110" "0010001" "0111101" "1000010" "1001111" "0011000" "1000111" "0111000"	'1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0' '1' '0'

➔ **Afficheur 7 segment :**

Il place chaque segment dans l'afficheur qui lui convient ce selon la case donnée par l'entrée af.

- **la partie opérative :**

L'assemblage de tous les sous blocks nous donnes :



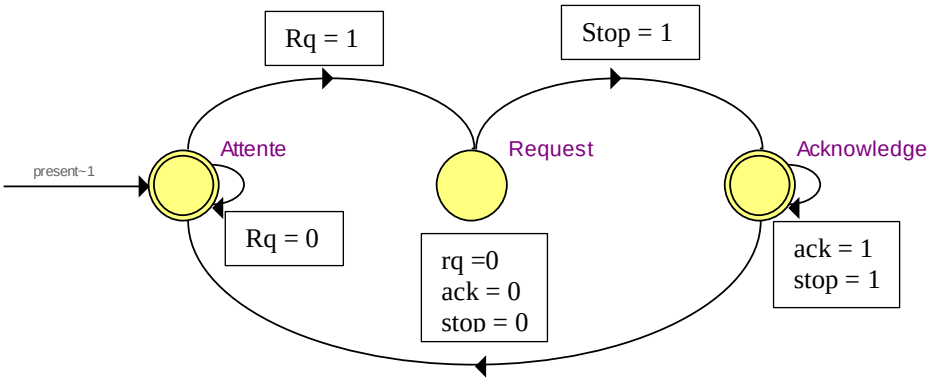
- **Partie commande (partie contrôle) :**

La machine d'état gère 3 signaux :

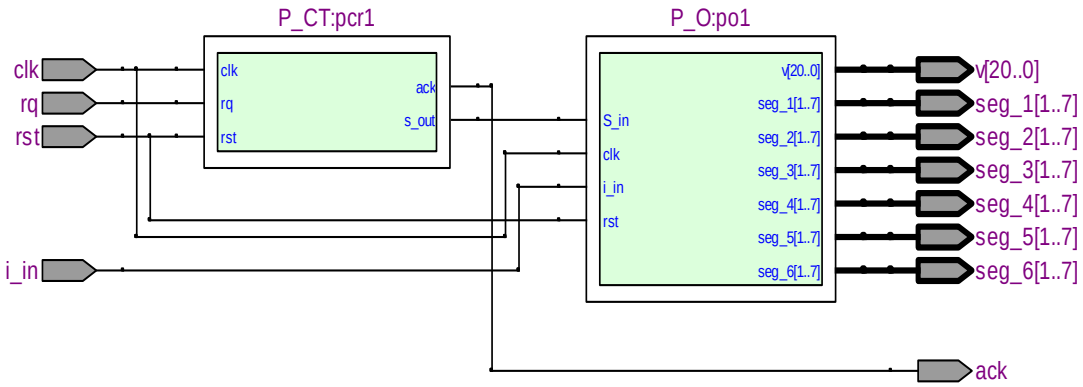
- **Request :** Entrée indiquant au composant qu'une valeur veut être lue.
- **Acknowledge :** Sortie indiquant au système externe qu'une valeur peut être lue .
- **Stop :** Signal interne permettant d'assurer la stabilité de la sortie V lors de sa lecture (actif haut).

Pour effectuer la gestion du protocole de communication, nous avons besoin de 3 états :

- **Attente :** C'est l'état initial, c'est à dire qu'il n'y a aucune requête en cours, donc le signal Acknowledge est à 0. De plus lorsque l'on est dans cet état, la valeur de V peut changer, donc le signal Stop est à 0. Lorsqu'une requête est détectée, la machine d'état passe dans l'état suivant Request.
- **Request :** C'est l'état qui met en place les conditions nécessaires à la lecture de V par le système externe, c'est à dire que l'on garantit la stabilité de V en mettant Stop à 1. La ligne Acknowlege reste 0 pour laisser le temps à V de se stabiliser. La machine d'état passe ensuite systématiquement à l'état suivant, Acknowledge.
- **Acknowledge :** La machine d'état continue à assurer la stabilité de V en gardant Stop à 1 et la ligne Acknowledge passe à 1 pour indiquer que la lecture est possible. Le passage à l'état suivant se fait lors de la détection du passage de la ligne Request à 0 ce qui indique la fin de la lecture. La machine d'état retourne donc dans l'état Attente.



- **L'architecture Top du circuit :**



b- Simulation et teste du circuit :

Pour la simulation est le teste on a considéré que l'horloge a une fréquence de 25 MHz et le capteur d'détecte une impulsion chaque 50 ns la longueur de l'impulsion aussi est égale à 50 ns.

➔ Premier teste :

La ligne I représente les impulsions envoyées par le moteur, v est la sortie du composant représentant la vitesse. Vers 3650 ns le moteur a effectué son premier tour car le compteur d'impulsion arrive à 29. cmpt30 passe donc à 1 pendant un cycle d'horloge afin d'indiquer qu'il faut afficher la nouvelle valeur de v. v_int qui sauvegarde la valeur de la vitesse est mis à la nouvelle valeur de la vitesse, c'est à dire la valeur du compteur v_temp. Au coup d'horloge suivant V affiche la nouvelle valeur de la vitesse qui est 91 ns dans notre cas, en vois aussi la sortie de v de l'afficheur 7 segment. Les compteurs de vitesse et d'impulsions sont remis à zéro. On peut vérifier que le compteur modulo 30 est bien incrémenté à chaque impulsion I. De même le compteur qui indique la vitesse par tour est compte à chaque front montant de l'horloge.

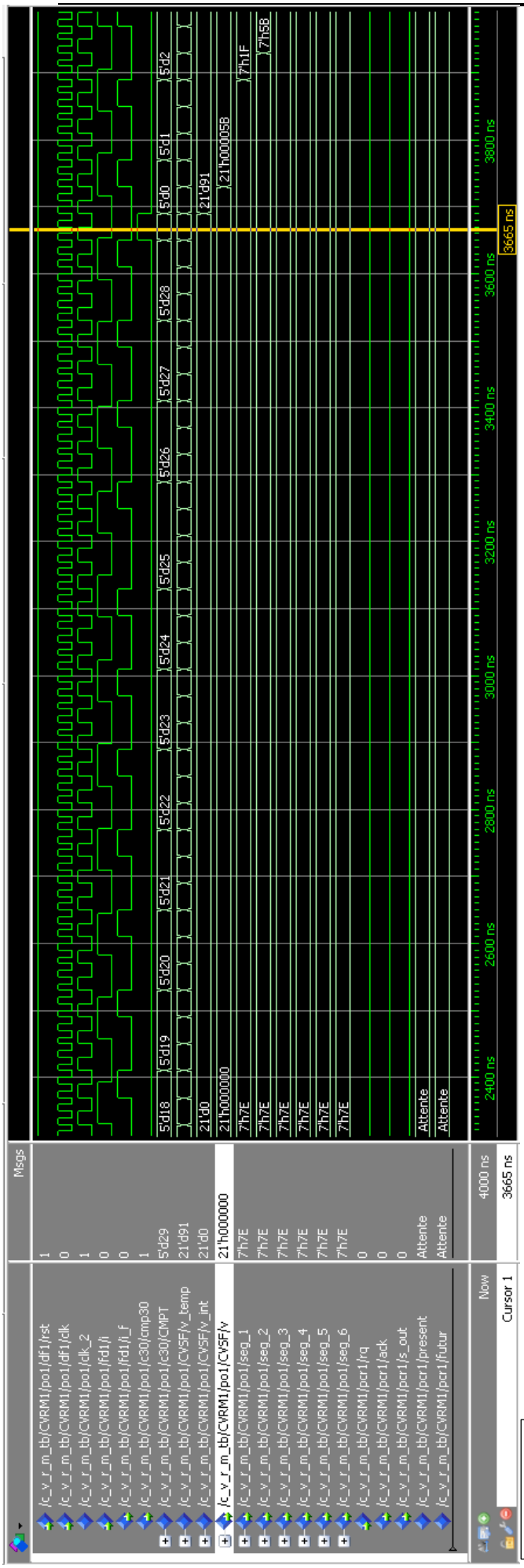
Pendant ce temps la machine d'état est restée à l'état d'attente car il n'y a pas eu de demande de lecture. On simule à 4000 ns un Request du système extérieur. Deux coups d'horloge plus tard (le temps de certifier que V est fixe en passant par l'état Request de notre machine d'état), ack passe à 1 pour indiquer qu'une lecture est possible. Lorsque la lecture a été effectuée (rq passe à zéro), ack repasse à zéro pour indiquer au système extérieur qu'il peut à nouveau demander une lecture.

➔ Deuxième teste :

Pour vérifier que v est bien bloqué lorsqu'il y a une lecture on ré effectue une simulation mais en demandant une lecture (c'est à dire Request à 1) avant la fin du premier tour du moteur (passage de 0 à 91 de v). La valeur de v doit rester fixe et ne changer qu'à la fin du Request en prenant la nouvelle valeur. L'étage de sauvegarde de la partie logique n'est utile que dans cette éventualité.

On remarque effectivement que sur le chronogramme, à la fin du premier tour du moteur vers 3650 ns, V ne change pas jusqu'à 3850 ns. En effet sa valeur est garantie stable car il y a une lecture en cours. Stop est à 1 et bloque le changement de v. On peut observer que v_int a pris la nouvelle valeur de la vitesse mais qu'elle n'est pas affectée à V. Dès que la lecture est terminée, Stop repasse à 0 et v prend la nouvelle valeur de la vitesse qui avait été sauvegardée dans v_int. Le fonctionnement est conforme à nos attentes.

Teste 1 :



Teste 2 :

CONCLUSION :

D'après la réalisation de ce projet, on a pu aboutir à apporter une réponse à notre problématique qui consiste à créer **« un calculateur de vitesse de rotation d'un moteur »** en utilisant Modelsim PE Student Edition en utilisant le langage VHDL. Ce par la conception et la simulation des circuits de base (multiplexeur, registre, comparateur, …), puis la conception et la simulation des sous blocks de la partie opérative qui a engendré la facilitation de l'assemble de la partie opérative et sa simulation. Ensuite la création de la partie contrôle et sa simulation. Et à la fin l'aboutissement à la conception et la simulation de l'architecture TOP du système. Après l'étape de la conception et la simulation, on a passé à l'étape de synthèse avec le logiciel QUARTUS II 9.0 sur la carte FPGA EP2C70F896C6

INDEX :

https://zestedesavoir.com/tutoriels/371/les-compteurs-numeriques/

https://www.enib.fr/~kerhoas/vhdl_lab1.html

https://www.google.com/url?sa=t&source=web&rct=j&url=http://www.uqac.ca/ht2bui/compteur_vhdl.pdf&ved=2ahUKEwit2oP7mdjrAhVD3aQKHeRoB34QFjAAegQIAhAB&usg=AOvVaw35ySB3fxSEJ0PCmHFOJsI-

https://www.hella.com/techworld/fr/Technique/Capteurs-et-Actionneurs/Capteur-ABS-vitesse-roue-depannage-4074/

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj0-Ou3sNzrAhV6QRUIHStqDpUQFjAAegQIBBAB&url=http%3A%2F%2Fwww.brodeurelectronique.com%2Fprojets%2FprojetVHDL%2FprojetVHDL.pdf&usg=AOvVaw3JFJDWXH7LhsfGT28B5POx

http://nec-itplatform.fr/quest-ce-quun-diviseur-de-frequence/

➔ Des circuits vus dans les TPs.